MLP Coursework 3: Project Interim Report

Laszlo Treszkai, and a student who wished to remain anonymous (equal contribution)

Abstract

Literature has shown that modern neural networks can give a poor estimation of their uncertainty, but well-calibrated models are important in many applications, such as those responsible for human safety. In this project we investigate the effect of various factors on model calibration. This interim project report details our baseline experiments, where we treat the softmax output of a fully connected three-layer network as its confidence in the prediction. Our experiments showed that our baseline model is already well-calibrated when trained on the EMNIST By-Class dataset. Calibration worsened when we used only a subset of the training set. We experimented with various regularisation constants for weight decay, and found that increasing regularisation increases calibration, but too much regularisation leads to a decrease in both accuracy and calibration. One of our main findings is that cross-entropy error is not a good indicator of model calibration. We also describe our plans for the next four work weeks, where we intend to perform similar experiments with more advanced calibration methods such as deep ensembles and test-time dropout methods.

1. Introduction

Deep neural networks have seen significant improvements in accuracy following recent advances (Goodfellow et al., 2016), with state-of-the-art performance achieved in various fields including computer vision (Krizhevsky et al., 2012) and speech recognition (Hinton et al., 2012). While the predictions of these models are highly accurate, for many applications it is desirable or even critical for our models to indicate when its predictions are uncertain and to what degree. For example, a speech recognition system would ideally indicate its uncertainty so that human annotators can check only those parts where the system has the lowest confidence. In autonomous driving systems, misclassifications with high certainty might contribute to fatal incidents.

In this work we measure the effect of different factors on model calibration in a supervised image classification task using the EMNIST By-Class dataset. First, we check if expected calibration error (ECE) changes qualitatively similarly as prediction error and cross-entropy error (which in this case is the same as negative log-likelihood, NLL). Then we change the number of training samples, and finally add weight decay to the model with different regularisation constants. The effects on calibration are evaluated, and our results are compared against those of Guo et al. (2017).

In this interim report we describe the baseline experiments we performed, where we treat the softmax outputs of a single fully-connected neural network as a measure of its certainty. For our final report, we plan to implement methods that use dropout at test time: MC dropout (Gal & Ghahramani, 2016), which can be interpreted as approximate inference in a Bayesian interpretation of the network, and Concrete Dropout (Gal et al., 2017), which was developed to circumvent the need to do a grid search over dropout probabilities when doing MC dropout. These methods are to be compared against our baseline results, to see if they show similar patterns in the aforementioned questions and if they really improve calibration.

In the following section we formalise the problem of model calibration and different ways to evaluate it, namely calibration plots and expected calibration error. Section 3 describes the system used in our experiments, including the network architecture and uncertainty estimation method. Section 4 gives details on the performed experiments, which show how model calibration changes during training, and also plots calibration as a function of training set size and weight decay constant. Finally, Section 5 summarises our interim conclusions, in Section 6 we describe our plans for coursework 4.

2. Description of model calibration

In this report we focus on the calibration of supervised multi-class classifiers, where the dataset contains inputs $\mathbf{x} \in \mathcal{X}$ together with their true labels $y \in \mathcal{Y}$. The dataset contains independent samples from a groun distribution $\pi(\mathbf{x}, y)$. Given an input $\mathbf{x}$, a softmax classifier outputs a vector $\hat{\mathbf{p}}$. Ideally, we want $\hat{p}_i$ to be equal to the probability that the input is in class i. If this always holds, then we say that the model is *perfectly calibrated*. For example, if the model makes 1000 predictions for independent test samples, each with confidence $\hat{p} := \operatorname{argmax}(\hat{\mathbf{p}}) = 0.7$, then approximately 700 of them should be correct. Formally, this can be expressed with the following equation:

$$\mathbb{P}_{\pi(\mathbf{x},y)}(\hat{y} = y | \mathbf{x}, \hat{p}) = \hat{p}. \qquad (1)$$

Usually $\mathcal{X}$ is a high-dimensional continuous space (e.g. in the case of EMNIST, the inputs come from the 784-dimensional unit cube), and the distribution $\pi(\mathbf{x}, y)$ is unknown, so we cannot calculate this probability exactly. Below we describe two methods to evaluate model calibration.

Qualitative evaluation. We can represent the model accuracy as a function of its certainty on a *calibration plot*, also known as a *reliability diagram* (DeGroot et al., 1982; Niculescu-Mizil & Caruana, 2005) – Figure 2 shows an example. The variable $\hat{p} \in [0, 1]$ is a continuous variable, so the interval $[0, 1]$ is first split into M equal-width disjoint intervals $I_m = (\frac{m-1}{M}, \frac{m}{M}]$. The examples are then split into M bins B_m $(m = 1 \dots M)$, where for each $i \in B_m$, $\hat{p}_i \in I_m$. The accuracy of a bin is defined as usual:

$$\mathrm{acc}(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} \delta_{y_i, \hat{y}_i}, \tag{2}$$

where δ is the Kronecker delta. The mean confidence of a bin is defined as follows:

$$\mathrm{conf}(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} \hat{p}_i. \tag{3}$$

These numbers are plotted on a confidence vs. accuracy chart. If and only if the model is perfectly calibrated, $\mathrm{acc}(B_m) = \mathrm{conf}(B_m)$ for all $1 \leq m \leq M$. Note that we cannot judge the calibration of the model from only the calibration plot, as it doesn't show the number of examples in each bin. For this reason, we always include the histogram of $\hat{p}$ along with the calibration plots.

Quantitative evaluation. From a calibration plot we can judge if a model is miscalibrated, but when comparing a large number of models, it is more convenient to express the level of calibration with a single number. *Expected calibration error* is the average difference between the accuracy and confidence of the binned predictions: $\mathrm{acc}(B_m)$ and $\mathrm{conf}(B_m)$ (Naeini et al., 2015) :

$$ECE = \frac{1}{N} \sum_{m=1}^{M} |B_m| \cdot \left| \mathrm{acc}(B_m) - \mathrm{conf}(B_m) \right|, \tag{4}$$

where $N = \sum_m B_m$ is the number of samples. ECE is zero if and only if the model is perfectly calibrated; and in the worst case $ECE = 1$, or 100%. Note that perfect calibration does not imply perfect accuracy: when predicting class labels, the model that always outputs the prior class probabilities is perfectly calibrated, but its accuracy is only as big as the proportion of the largest class, which is usually far from 100%.

3. Baseline systems

The baseline system was a neural network with 2 fully connected hidden layers (with 512 hidden units each), ReLU activation functions, trained to minimize cross-entropy softmax error. The weights were initialised with Glorot uniform initialisation (Glorot & Bengio, 2010), and the bias vectors initialised with zeros. In Coursework 2 we saw that on the EMNIST task such a network architecture leads to almost as good performance as deeper networks or convolutional networks (s1765864 (2017) found a test set accuracy of 84.5% with this architecture vs. 88.3% with convolutional nets), while it can be trained efficiently, which motivated our decision to use it as a baseline system.

While more complex methods can achieve much better performance, our goal in this report is to investigate model calibration, not to increase model accuracy. Furthermore, fully connected networks are still frequently used in prediction tasks (Kuleshov & Liang, 2015), so our results could lead to improve best practices in fields where well-calibrated systems are preferred over highly accurate ones.

Apart from the experiments in section 4.5 (where we explicitly noted the regularisation constant), the models were trained without regularisation.

The models were trained with Adam learning rule (Kingma & Ba, 2014), with the default settings of Keras (learning rate 0.001, $\beta_1 = 0.9$, $\beta_2 = 0.999$). In section 4.3 we used stochastic gradient descent (SGD) (Robbins & Monro, 1951) with a low learning rate, in order to deliberately increase the training time, so that the evolution of error and calibration can be assessed more accurately. All experiments used a minibatch size of 100.

As a baseline method (i.e. in every reported experiment), we treat the maximum of the softmax output as the confidence of the model, $\hat{p}$. Guo et al. (2017) found that these values are not well calibrated on the CIFAR-100 dataset with more complex networks ($ECE = 4.9\% \dots 16.5\%$), while the same network architectures had slightly better calibration on CIFAR-10 ($ECE = 3.0\% \dots 4.5\%$). We experienced similar differences between MNIST and EMNIST datasets.

4. Experiments

4.1. Baseline on MNIST

We briefly experimented with the MNIST dataset (LeCun, 1998), and found that our model achieves an accuracy of 98.2%, with the vast majority of the predictions made with over 98% confidence.

The confidence and accuracy figures were high enough that calibration was mostly irrelevant – the model was almost perfectly calibrated. Hence we decided to focus on the EMNIST By-Class dataset (Cohen et al., 2017) for the remainder of our experiments, so that we have a more challenging learning task where calibration can be a problem.

The EMNIST By-Class dataset contains 697,932 images of handwritten characters from 62 classes, corresponding to the numerical digits, uppercase letters and lowercase letters. In contrast to the EMNIST Balanced and By-Merge datasets, pairs of easily confusable characters are not merged in the EMNIST By-Class dataset. Alongside the imbalance of the dataset, this makes for a more challenging classification problem. The distribution of our training set is shown in Figure 1. Apart from section 4.4, We used a 50% – 50% split for the training and validation sets.

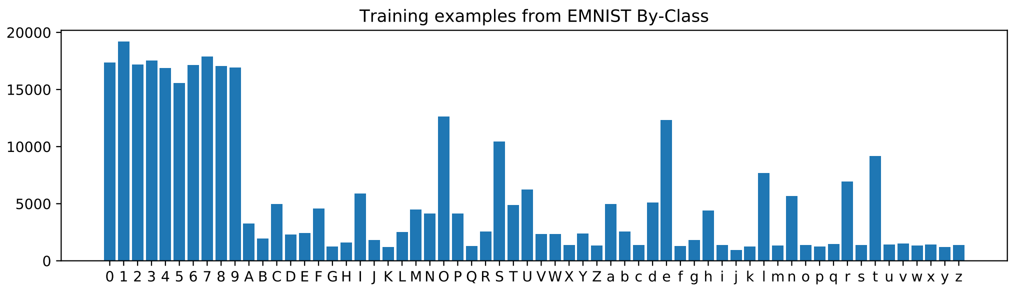

Figure 1. Distribution of the training examples we took from the EMNIST By-Class dataset.

4.2. Baseline on EMNIST

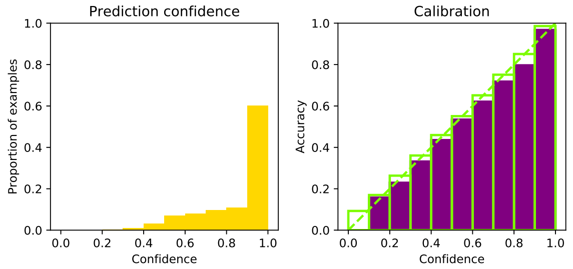

Figure 2. Baseline network trained on EMNIST By-Class. Left: Distribution of prediction confidence. Right: Calibration plot. (The green bins represent a perfectly calibrated model.) Both were evaluated on the validation set.

On the EMNIST By-Class dataset, our baseline model achieved 84.2% classification accuracy. Compared to MNIST, the distribution of prediction confidences was much more interesting, with an average of 86.7%. The model also turned out to be reasonably well-calibrated, with an expected calibration error (ECE) of 2.5%. Figure 2 gives confidence and calibration plots for the model.

It has been shown that increasing model capacity leads to miscalibration (Guo et al., 2017), so the fairly low ratio of network parameters to training examples could be a reason why our baseline is well-calibrated. (The model had 696,382 parameters, and we trained on 348,966 examples).

4.3. SGD with low learning rate

After testing out the baseline on EMNIST By-Class, we investigated how calibration evolves during training. Earlier, when we had Adam as the optimiser, the model completed training within 6 epochs. For clearer results in this section, we prolonged the training process by using SGD with a learning rate of 0.01.

It turned out that 0.01 was a low enough learning rate that our baseline model did not overfit within 100 epochs: the classification errors over the validation set are shown in Figure 3.

We observed that calibration deteriorated as classification accuracy improved. We can see from Figure 3 that past epoch 40, classification error continues declining (albeit noisily) while ECE increases drastically. In other words, the network learned to increase its classification accuracy at the expense of well-calibrated predictions.

A similar observation was also made in (Guo et al., 2017, Section 3), which applied a 110-layer ResNet to the CIFAR-100 dataset. To see the result visually, the paper used NLL as an indirect measure of calibration. However, our results in Figure 3 suggest that NLL might be a poor measure of calibration in this situation, since NLL and ECE became strongly disconnected after epoch 40. We would argue that comparing ECE with classification error is more sensible, since ECE measures calibration more directly.

4.4. Varying the training set size

Next we studied how decreasing the training set size affects the calibration of our baseline model. We tried 12 different sizes from 50,000 to 300,000, each with 5 different random seeds. The random seed affects the network weight initialisation as well as the shuffling of training data at each epoch.

The results are shown in Figure 4. Naturally, classification accuracy dropped as the training set became smaller. At the same time, calibration suffers as we can see from the ECEs. We notice that model calibration is sensitive to changes to the random seed, percentage-wise to a larger extent than the classification accuracy.

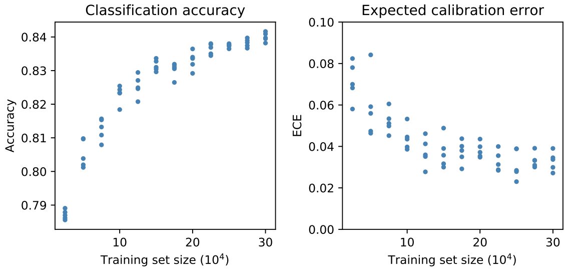

Figure 4. Classification accuracy and ECE for different training set sizes. For each training set size, we trained one baseline network for each of 5 different random seeds.

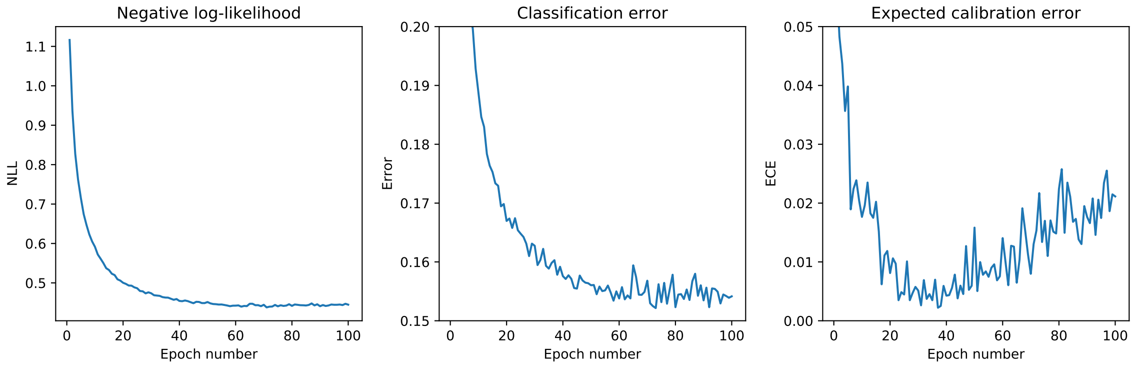

Figure 3. Various metrics evaluated over the validation set for our baseline network trained on EMNIST By-Class. For this experiment, the optimiser was SGD with a learning rate of 0.01.

4.5. Weight decay

Weight decay is often used as a form of regularisation in neural networks, as a means to increase the model's ability to generalise.

Experiments in literature showed that changing the regularisation constant can improve model calibration significantly (Guo et al., 2017). We checked if this holds for our baseline network by applying L2 regularisation with different regularisation constants, ranging from $\lambda = 10^{-7}$ to 10^{-2}. The results are shown in Figure 5.

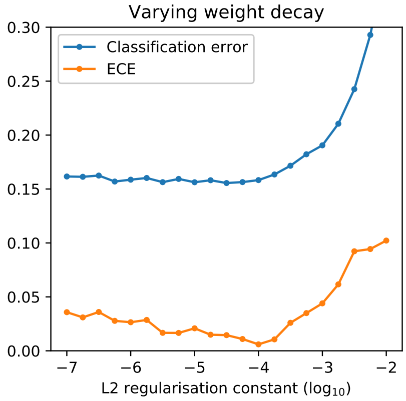

Figure 5. Classification error and ECE for various regularisation constants, applied to our baseline network.

While classification accuracy stayed the same with λ between 10^{-7} and 10^{-4}, in this range model calibration improved with increasing regularisation. (Error changed from 83.9% to 84.2%, while ECE decreased from 3.1% to 0.6%.)

At $\lambda = 10^{-4}$, classification accuracy started to deteriorate, and at the same time calibration worsened at a much steeper rate.

Guo et al. (2017, Section 3) found that model calibration can improve by increasing the weight decay constant, well after the model achieves minimum classification accuracy. This was not the case for our baseline, as we can see from Figure 5.

This shows that the combination of network architecture and dataset affects the relationship between classification error and calibration, and drawing general conclusions from the results of an experiment on one dataset with one model is premature.

5. Interim conclusions

Our experiments showed that our baseline model consisting of two fully-connected hidden layers and a softmax output layer was reasonably well-calibrated.

We discovered experimentally that varying the training set size or model hyperparameters such as weight decay can contribute to miscalibration. Calibration worsened as we decreased the number of training samples. The regularisation constant had a single optimum that led to best calibration, and deviating from it either way led to worse calibration error, which agrees with the findings of Guo et al. (2017). In one experiment, we observed that our baseline network learned to improve its classification accuracy at the cost of calibration after a certain epoch. The same experiment demonstrated that NLL and ECE do not move hand-in-hand as training progresses, despite the use of NLL as the measure of calibration by some authors (Lakshminarayanan et al., 2016).

6. Future work

Most of the experiments we have performed so far were intended to provide a baseline on which we could evaluate methods for improving calibration such as ensembling and test-time dropout. In the light of our discovery that our baseline network was fairly well-calibrated, we turned to exploring various factors which could affect calibration, and managed to investigate the effect of regularisation and training set size.

Our main intention now is to implement ensembling (Lakshminarayanan et al., 2016), MC dropout and Concrete Dropout (Gal et al., 2017), and see if they demonstrate the patterns we have seen from varying hyperparameters. We plan to complete the implementations and the majority of experiments within the next two weeks (calendar weeks 9 and 10), and devote a week left to writing the report. The final week is reserved as slack time, where we can perform further investigations if the project goes according to plan.

It could well turn out that ensembling and test-time dropout do not make a significant impact on calibration. In this case, there might still be interesting results from varying hyperparameters, similar to those we have observed so far. A risk with our plan is that test-time dropout, especially Concrete Dropout, might take more effort to implement than we can afford. To mitigate this risk, we have located a reference implementation for Concrete Dropout written for Keras, which can be dropped into our experimental framework if necessary.

References

Cohen, Gregory, Afshar, Saeed, Tapson, Jonathan, and van Schaik, André. EMNIST: an extension of MNIST to handwritten letters. *CoRR*, abs/1702.05373, 2017. URL http://arxiv.org/abs/1702.05373.

DeGroot, Morris H, Fienberg, Stephen E, and of Statistics, Carnegie-Mellon University. Department. The comparison and evaluation of forecasters. 1982. Includes abstract.

Gal, Y., Hron, J., and Kendall, A. Concrete Dropout. *ArXiv e-prints*, May 2017.

Gal, Yarin and Ghahramani, Zoubin. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pp. 1050–1059, 2016.

Glorot, Xavier and Bengio, Yoshua. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pp. 249–256, 2010.

Goodfellow, Ian, Bengio, Yoshua, and Courville, Aaron. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. On Calibration of Modern Neural Networks. *ArXiv e-prints*, June 2017.

Hinton, Geoffrey, Deng, Li, Yu, Dong, Dahl, George, Mohamed, Abdel-rahman, Jaitly, Navdeep, Senior, Andrew, Vanhoucke, Vincent, Nguyen, Patrick, Sainath, Tara, et al. Deep neural networks for acoustic modeling. 2012.

Kingma, Diederik P. and Ba, Jimmy. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. URL http://arxiv.org/abs/1412.6980.

Krizhevsky, Alex, Sutskever, Ilya, and Hinton, Geoffrey E. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097–1105, 2012.

Kuleshov, Volodymyr and Liang, Percy. Calibrated structured prediction. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'15, pp. 3474–3482, Cambridge, MA, USA, 2015. MIT Press. URL http://dl.acm.org/citation.cfm?id=2969442.2969627.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles. *ArXiv e-prints*, December 2016.

LeCun, Yann. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

Naeini, Mahdi Pakdaman, Cooper, Gregory F., and Hauskrecht, Milos. Obtaining well calibrated probabilities using bayesian binning. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, pp. 2901–2907. AAAI Press, 2015. ISBN 0-262-51129-0. URL http://dl.acm.org/citation.cfm?id=2888116.2888120.

Niculescu-Mizil, Alexandru and Caruana, Rich. Predicting good probabilities with supervised learning. In *Proceedings of the 22Nd International Conference on Machine Learning*, ICML '05, pp. 625–632, New York, NY, USA, 2005. ACM. ISBN 1-59593-180-5. doi: 10.1145/1102351.1102430. URL http://doi.acm.org/10.1145/1102351.1102430.

Robbins, Herbert and Monro, Sutton. A stochastic approximation method. *Ann. Math. Statist.*, 22(3):400–407, 09 1951. doi: 10.1214/aoms/1177729586. URL https://doi.org/10.1214/aoms/1177729586.

s1765864. MLP Coursework 2: Learning rules, BatchNorm, and ConvNets, 2017.